

Stable haptic interaction based on adaptive hierarchical shape matching

Yuan Tian¹ (✉), Yin Yang², Xiaohu Guo¹, and Balakrishnan Prabhakaran¹

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract In this paper, we present a framework allowing users to interact with geometrically complex 3D deformable objects using (multiple) haptic devices based on an extended shape matching approach. There are two major challenges for haptic-enabled interaction using the shape matching method. The first is how to obtain a rapid deformation propagation when a large number of shape matching clusters exist. The second is how to robustly handle the collision response when the haptic interaction point hits the particle-sampled deformable volume. Our framework extends existing multi-resolution shape matching methods, providing an improved energy convergence rate. This is achieved by using adaptive integration strategies to avoid insignificant shape matching iterations during the simulation. Furthermore, we present a new mechanism called stable constraint particle coupling which ensures consistent deformable behavior during haptic interaction. As demonstrated in our experimental results, the proposed method provides natural and smooth haptic rendering as well as efficient yet stable deformable simulation of complex models in real time.

Keywords deformation; haptic rendering; shape matching; multi-resolution

1 Introduction

Haptic-enabled deformable simulation has been an important research topic. Haptic devices provide

enhanced computer–human interaction in which the user is not only able to visually observe 3D geometric changes in an object, but also feel and touch the object in the virtual world. It plays a crucial role in various applications such as computer animation [1, 2], medical training [3–6], virtual rehabilitation [7], etc. Many computational methods, such as the finite element method (FEM) or mass–spring systems, have been developed to model the dynamic 3D volumetric deformation of soft objects. The shape matching [8] technique is a competitive candidate for solving this problem as it has several important advantages for haptic-enabled deformable simulation. One of the most attractive is that it supports unconditionally stable integration under arbitrary user inputs (e.g., a large, abrupt, impulse-like force).

The shape matching method typically subdivides the volume of the deformable object into many overlapping subgroups or *clusters* of particles. Local as-rigid-as-possible transformation of a cluster is computed first, while the global deformation is obtained after local displacement information for clusters has been sufficiently exchanged. Geometrically complex 3D models usually have a large number of clusters to capture detailed local deformation, which can lead to slow deformation convergence, especially considering that haptic devices often require high frame rates in practical applications. The situation becomes more complicated with the involvement of multiple haptic devices. Because the classical shape matching approach is based on a position-driven pseudo-dynamic system, some fundamental physical parameters and relations are not clearly defined within this framework. Robustly yet efficiently accommodating the interaction between the haptic

1 University of Texas at Dallas, Richardson, TX 75080, USA. E-mail: Y. Tian, yuan.tian1@utdallas.edu (✉); X. Guo, xguo@utdallas.edu; B. Prabhakaran, praba@utdallas.edu.

2 University of New Mexico, Albuquerque, NM 87131, USA. E-mail: yangy@unm.edu.

Manuscript received: 2015-08-27; accepted: 2015-08-29

interaction points (HIPs) and the deformable object is another key challenge.

To address the aforementioned issues, we propose a novel haptic-enabled framework using a particle cluster hierarchy. The proposed framework is able to effectively boost the convergence rate of the deformation energy during the simulation via three adaptive iteration strategies which track the variation of energy density during the simulation and thereby avoid unnecessary computation. The key component of the framework is a new mechanism that ensures smooth interaction between the haptic devices and the virtual deformable objects. The deformation trajectory remains consistent after an HIP hits the object. This method intuitively supports stable and efficient force rendering with multiple haptic devices. Specifically, the technical contributions of this work can be summarized as follows:

- We present an efficient and robust framework that allows real-time interaction with 3D deformable objects using (multiple) haptic devices. It inherits the unconditional stability and high efficiency of the shape matching approach to provide natural and smooth haptic rendering.
- Based on a cluster hierarchy, an optimized adaptive simulation algorithm is given to accelerate energy convergence, allowing the capture of rich local deformation details.
- A new technique, *stable constraint particle coupling*, is provided for haptic interaction. When an HIP collides with the deformable object, the original optimal as-rigid-as-possible status of the object is not influenced by the newly inserted particle, eliminating jittering artifacts.

2 Related work

Simulating 3D elastic deformation using FEM has long been an active topic in many research communities including computational mechanics, computer graphics, and virtual reality. Pentland and Williams [9] borrowed the idea of *modal analysis* that decomposes the 3D deformation into vibrations of different frequencies. The computation speed is greatly boosted by discarding the high-

frequency modes. This method was later extended to co-rotational deformable models [10], nonlinear deformation modes [11], and hybrid deformable models [12]. It has been successfully adopted for real-time haptic interaction [1, 13–15]. An *invertible element* [16] approach provides robust FEM simulation under extreme deformations. However, because of its high computational cost, this method is impractical for direct deployment for haptic interaction.

Another line of contributions uses a simplified physical model and constructs the simulator with a more intuitive formulation. For example, the mass–spring system approach [17, 18] adopts spring-connected mass particles to model the force–displacement relation. A mass–spring system strongly couples each pair of neighboring particles, requiring the iterative solution of a large nonlinear system at each time step. While recent research has significantly improved the efficiency of integration of the mass–spring system [19], it is still quite challenging to handle geometrically complex models in real-time with haptic devices. However, the shape matching method, using *position based dynamics* (PBD) [20, 21], is able to provide a fast, controllable, and unconditionally-stable dynamic simulation. Unlike mass–spring systems, this method [8, 22] is essentially a meshless method grouping the particle cloud into clusters. The computation associated with each cluster is independent making the shape matching based deformable model much more lightweight. Rivers and James [23] used overlapped clusters (in a lattice) to control the stiffness of the deformable object. Steinemann et al. [24] extended this work by using dynamic adaptive selection of levels of details (LODs). A similar idea has also been applied for quasi-static mesh deformation [25, 26].

In such cluster-based shape matching methods, the local optimum of the deformation energy is found by computing the best fitting rigid body rotation and translation. Each cluster has no information about its neighbors until the average displacement of each overlapping region is determined. The system has to repeat this matching and averaging procedure many times in order to sufficiently reduce the energy for a large deformation, similarly to the well-known Jacobi solver. To improve the energy convergence rate, multigrid methods [27–29] have been adopted. In

this paper, we extend the idea of hierarchical shape matching (HSM) with enhanced adaptive strategies to further reduce the cost introduced by the cluster hierarchy, which can be seamlessly integrated within state-of-the-art shape matching based frameworks.

Haptic devices typically require high-rate force rendering in order to deliver a satisfactory user experience. Accordingly, Otaduy and Lin [30] proposed a multi-rate architecture splitting the haptic rendering pipeline into a haptic thread and a contact thread. Here, force rendering is independent of the time integration of the dynamic system. As shown in Fig. 1, we also adopt this framework, running the haptic thread and the simulation thread asynchronously. The responsive force to be rendered via haptic devices is often approximated as a spring-like force [3, 31] based on the Euclidean distance between the contact point and the HIP. Instead, we use a more physically meaningful approach that formulates the collision force as the partial derivative of the deformation energy [1, 2, 32], following the intuition that forces are always dragging the deformed particles back to their rest positions. Virtual force coupling [33–35] is also used in our system to ensure that the haptically rendered force is smooth and natural.

3 Adaptive shape matching using cluster hierarchy

The surface geometry of the deformable object is represented by a triangle mesh. A volumetric particle cloud is automatically generated by voxelizing the original triangle mesh. Each corner of a voxel (cube)

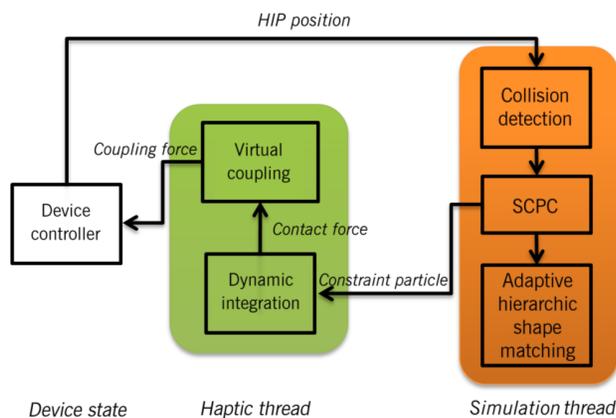


Fig. 1 Following existing work [30], our framework also uses a haptic thread and a simulation thread.

is associated with a particle as shown in Fig. 2. Particles are grouped into overlapping clusters. A natural choice is to select the eight particles of a voxel as a cluster. In this case, neighboring clusters sharing a facet have four overlapping particles. When the particle cloud deforms, the geometry of the embedded triangle mesh can be easily computed using trilinear interpolation. In the rest of the paper, we simply refer to the clustered particle cloud as the *cube mesh* or *cluster mesh*.

Shape matching of clustered particles. Each particle is associated with an initial position, a current position, and a goal position denoted respectively by \mathbf{x}_{ij}^0 , \mathbf{x}_{ij} , and \mathbf{g}_{ij} , for the j^{th} particle in the i^{th} cluster. Each particle is also assigned a mass m_{ij} . The goal position of the particles specifies a configuration in which the corresponding cluster has zero deformation potential. When external forces are applied, particles are virtually displaced via Newton’s second law without consideration of internal forces at first. The goal positions of each cluster’s particles represent a certain rigid body motion (i.e., a null-deformation displacement) that is closest to the displaced cluster. Thus, the goal position of particle j in cluster i is given by

$$\mathbf{g}_{ij} = \mathbf{R}_i \mathbf{x}_{ij}^0 + \mathbf{t}_i \tag{1}$$

where $\mathbf{R}_i \in \text{SO}^3$ and $\mathbf{t}_i \in \mathbb{R}^3$ represent the best fitting rotation and a translation to be determined, respectively. It can be shown that \mathbf{t}_i is just the offset of the cluster centroid and \mathbf{R} can be computed by applying polar decomposition or SVD to the moment matrix of the cluster. We refer to such computation for obtaining \mathbf{R} and \mathbf{t} for each cluster as *shape matching* (SM). The quadratic deformation energy or potential E_i is defined as the mass-weighted sum of the square distances between the current positions and the goal positions for all particles in cluster i :

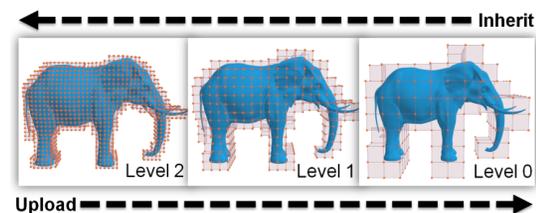


Fig. 2 Elephant model and corresponding cube meshes of different resolutions: the coarse cube mesh is generated by voxelizing the bounding volume of the surface mesh at resolution $10 \times 10 \times 10$. Finer cube meshes are generated by subdivision.

$$\begin{aligned}
 E_i &= \sum_j m_{ij} | \mathbf{g}_{ij} - \mathbf{x}_{ij} |^2 \\
 &= \sum_j m_{ij} | \mathbf{R}_i \mathbf{x}_{ij}^0 + \mathbf{t}_i - \mathbf{x}_{ij} |^2 \quad (2)
 \end{aligned}$$

The displacements of overlapping particles in neighboring clusters are averaged so that the cube mesh does not split. Naturally, such displacement averaging impairs the optimality of \mathbf{R} and \mathbf{t} and one may need to repeat the SM computation multiple times. We call an individual iteration of performing SM and neighbor averaging an *SM iteration*. The procedure of SM iteration is similar to the so-called local–global optimization in recent contributions [36]. It is guaranteed that each iteration monotonically reduces the deformation potential of the entire voxel mesh. After SM iterations terminate, a forward Euler step is used, with time step size h , to update the velocity and displacement of all particles (subscripts omitted):

$$\begin{cases} \mathbf{v}(t+h) = \mathbf{v}(t) + \alpha(\mathbf{g}(t) - \mathbf{x}(t))/h + h\mathbf{f}_{\text{ext}}(t)/m \\ \mathbf{x}(t+h) = \mathbf{x}(t) + h\mathbf{v}(t+h) \end{cases} \quad (3)$$

Cluster hierarchy. Our method is inspired by the classic multigrid technique [37] and existing work [24, 27, 38] on using multi-resolution simulation to accelerate energy convergence. A cluster hierarchy of multiple levels is constructed. The coarsest cube mesh is initially built at the top level (level 0). Each cube is further divided into multiple sub-cubes at the next lower level as shown in Fig. 2. Sub-cubes outside the surface mesh are discarded. The resolution of the cube meshes as well as the depth of the hierarchy can be interactively specified in the GUI provided by our system. Typically, a hierarchy of three to five levels is used in our experiments. We may incorporate more overlapping particles, as used in fast lattice shape matching (FLSM) [23], to further adjust the “stiffness” of the object.

3.1 Adaptive hierarchical shape matching

Using the cluster hierarchy, the deformable simulation begins with the SM iteration at the top (coarsest) level. After sufficient energy reduction occurs, the algorithm proceeds to clusters at the next level, based on the results from the previous level. Afterwards, the external forces are incorporated using a forward Euler step (Eq. (3))

for clusters at the bottom level, which triggers vibrational deformations due to the inertia terms and pulls the clusters away from their goal positions. In the next time step, the initial configuration of top-level clusters (\mathbf{R} and \mathbf{t}) are set to the blended rotations (e.g., using Slerp [39]) and translations of lower-level clusters. This procedure mimics the standard *V-cycle* in the multigrid approach [37], and has been adopted in many existing SM-based frameworks [24, 27, 28]. We refer to this algorithm as *hierarchical shape matching* (HSM).

Our framework further improves the simulation efficiency. The key inspiration is that not all SM iterations play equally important roles in shaping the deformed geometry of the object. In fact, our experiment shows that a considerable number of SM iterations (over 25%) can be avoided by checking three conditions during the multi-level SM iteration: a termination condition, a subdivision condition, and an origination condition, which we now discuss in detail.

Termination condition. Following the intuition that the SM iteration goes to the finer level after current level iterations do not effectively reduce the energy potential, we evaluate the energy reduction rate r_l at the l^{th} level as

$$r_l = \sum_i \Psi_{l,i}^k - \Psi_{l,i}^{k-1} \quad (4)$$

where $\Psi_{l,i}$ defines the *energy density* of cluster i . Superscripts k and $k-1$ indicate the SM iteration index. $\Psi_{l,i}$ is computed as $\Psi_{l,i} = E_{l,i}/n_i d_l^2$, where n_i is the number of particles in the i^{th} cluster and d_l is the voxel size at level l . We use a threshold value T_r to examine the effectiveness of SM iteration. Iteration at the current level l is terminated and simulation moves to the next level when the *termination condition*, $C_t \equiv r_l < T_r$, is satisfied.

Subdivision condition. If clusters at coarser level already well capture the deformed mesh geometry, we should not perform iteration at finer levels. Thus, SM iteration should only be applied to clusters whose energy density is larger than some threshold, T_s , i.e., $C_s \equiv \Psi_{l,i} > T_s$. We call this the *subdivision condition*. All child clusters of a cluster C_s are called *active clusters*. Figure 3 shows how clusters evolve and SM information is passed as the Buddha model bends.

Origination condition. External forces cause

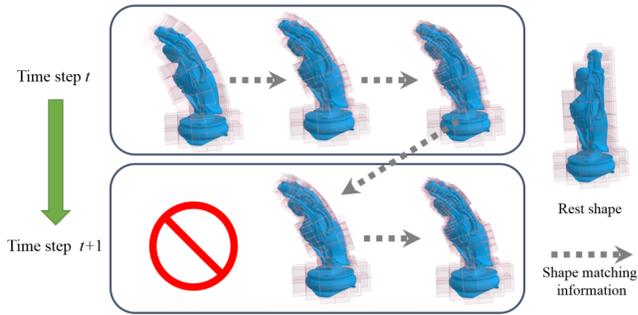


Fig. 3 Example of SM iteration strategy: at the t^{th} time step, simulation begins from the top level (level zero). Clusters in the top portion of the 3D model satisfy the subdivision condition and further SM iterations are continued at levels one and two. When the simulator advances to step $t + 1$, the origination condition at level zero fails. Therefore the simulation starts from level one. The dashed grey arrows indicate how SM information is passed.

the energy potential to increase. However it is possible that the external forces are subtle and local, and only cause small local deformations. For instance, a light breeze only sways the leaves of a willow tree while its major branches remain still (Fig. 10). Such small deviations of particles' positions will induce a larger perturbation of energy density for clusters at lower levels (because d_l is smaller). Blindly performing complete top-down SM iteration could include redundant computations and slow the simulation. Thus, we track the energy increase induced by forward Euler integration for all clusters subject to external forces¹ via $\sum_i \Psi_{l,i}^* - \Psi_{l,i}$, where $\Psi_{l,i}^*$ denotes the energy density after applying the forward Euler step. The *origination condition* finds a level whose energy reduction rate in the previous time step is comparable with the energy increase: $\mathcal{C}_o \equiv r_l \approx \sum_i \Psi_{l,i}^* - \Psi_{l,i}$, and sets it as the starting level.

Algorithm 1 outlines our adaptive SM iteration strategy. Our method differs from existing methods [24, 27, 28] by not only addressing the questions of “where and when should the iteration end?” but also “where and when should the iteration start?”, which further accelerates the energy convergence.

4 Stable constraint particle coupling

The core component in a haptic-enabled simulation

¹ Since forces are applied to particles at the bottom clusters, all of their parent clusters are also considered.

Algorithm 1: Adaptive shape matching

```

l ← 0; /* l is the current shape matching level */
while simulation is active do
  while l < n do
    for all clusters at level l do
      | inherit R and t from parents;
    end
    while !Ct do
      for all active clusters at level l do
        | shape matching iteration;
      end
      update rl;
    end
    for all active clusters at level l do
      if Cs then
        | set child clusters as active clusters;
      end
    end
    l ← l + 1;
  end
  /* now l = n */
  for all clusters do
    | forward Euler (Eq. (3));
  end
  update surface mesh;
  while l > 0 do
    if Co then
      | break; /* starting level of next step found */
    end
    else
      | l ← l - 1;
    end
  end
end
end
    
```

environment is a robust and efficient mechanism for handling the interaction between HIPs and the 3D deformable objects. While collision detection can be dealt with by most existing techniques [40], it is unlikely that the HIP really hits a particular particle inside the object. A simple solution to tackle this problem is to insert a new constraint particle (CP) into the cluster where the HIP resides. This method is referred to as *constraint particle coupling* (CPC) [2]. Unfortunately, adding CPs destroys the optimality of the computed \mathbf{R} and \mathbf{t} for the current SM iteration. Tian et al. [2] alleviate this problem by introducing an additional virtual particle or *ghost particle* (GP) paired with the CP so that the centroid of the cluster is maintained. However, jittery deformation still exists because such a method cannot guarantee to preserve the optimal rotation (Fig. 4). Motivated by this challenge, we propose an enhanced strategy called *stable constraint particle coupling* (SCPC), which ensures a smooth interaction between HIPs and objects.

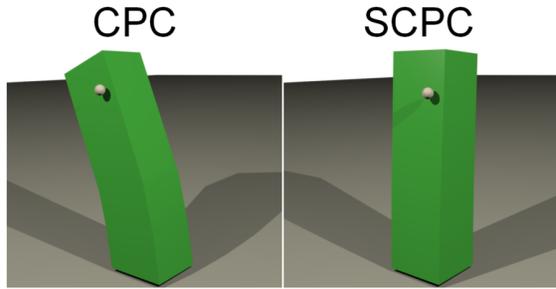


Fig. 4 Comparison between CPC and SCPC. An HIP hits the bar model along the normal direction of the front facet. CPC generates an unexpected lateral bending while SCPC leads to a more natural deformation response.

4.1 Optimal translation

Inserting a new CP corresponding to HIP yields a perturbation of the optimal translation $\mathbf{t} = \mathbf{c} - \mathbf{c}^0$, where \mathbf{c}^0 and \mathbf{c} denote the goal and current cluster centroid before the HIP hits the object, respectively. Following CPC [2], we insert an extra GP into the cluster; the updated cluster centroid can be written as

$$\begin{cases} \mathbf{c}^{0'} &= \frac{\sum m_j \mathbf{x}_j^0 + m_{cp} \mathbf{x}_{cp}^0 + m_{gp} \mathbf{x}_{gp}^0}{\sum m_j + m_{cp} + m_{gp}} \\ \mathbf{c}' &= \frac{\sum m_j \mathbf{x}_j + m_{cp} \mathbf{x}_{cp} + m_{gp} \mathbf{x}_{gp}}{\sum m_j + m_{cp} + m_{gp}} \end{cases} \quad (5)$$

Eliminating the perturbation of \mathbf{t} leads to

$$\mathbf{c} - \mathbf{c}^0 = \mathbf{c}' - \mathbf{c}^{0'} = \frac{\sum m_i \mathbf{x}_i}{\sum m_i} - \frac{\sum m_i \mathbf{x}_i^0}{\sum m_i} \quad (6)$$

Without loss of generality, we assume that the mass associated with each particle is the same and Eq. (6) leads to

$$\begin{cases} \mathbf{x}_{gp}^0 &= 2\mathbf{c}^0 - \mathbf{x}_{cp}^0 \\ \mathbf{x}_{gp} &= 2\mathbf{c} - \mathbf{x}_{cp} \end{cases} \quad (7)$$

which implies that the optimal translation can be maintained as long as the GP is inserted symmetrically with respect to the CP about the original cluster centroid.

4.2 Optimal rotation

One way to obtain the optimal rotation is to apply SVD to the moment matrix \mathbf{A} of the cluster, defined as

$$\mathbf{A} = \sum_j m_j \mathbf{p}_j \mathbf{q}_j^\top \quad (8)$$

where $\mathbf{q}_j = \mathbf{x}_j^0 - \mathbf{c}^0$ and $\mathbf{p}_j = \mathbf{x}_j - \mathbf{c}$ are the initial and current positions of the particle defined in a local coordinate frame with origin at the cluster centroid, respectively. Applying SVD to \mathbf{A} leads to

$\mathbf{A} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^\top$; the optimal rotation is just:

$$\mathbf{R} = \mathbf{U} \mathbf{V}^\top \quad (9)$$

Recall that in our implementation, each cluster has eight particles corresponding to the corners of a cube. Substituting Eq. (8) into the SVD formulation, following some manipulation, we obtain (again for convenience of explanation, assuming identical particle masses):

$$\mathbf{U}^\top (\mathbf{p}_1 \mathbf{q}_1^\top + \mathbf{p}_2 \mathbf{q}_2^\top + \dots + \mathbf{p}_8 \mathbf{q}_8^\top) \mathbf{V} = \mathbf{\Lambda} \quad (10)$$

Adding the CP as well as its paired GP adds two extra terms to the left-hand side of Eq. (10). Because of the centroid-symmetry relating the GP and CP ($\mathbf{p}_{cp} \mathbf{q}_{cp}^\top = \mathbf{p}_{gp} \mathbf{q}_{gp}^\top$), the left of Eq. (10) becomes

$$\mathbf{U}^\top (\mathbf{p}_1 \mathbf{q}_1^\top + \dots + \mathbf{p}_8 \mathbf{q}_8^\top) \mathbf{V} + 2\mathbf{U}^\top (\mathbf{p}_9 \mathbf{q}_9^\top) \mathbf{V}$$

Keeping both \mathbf{U} and \mathbf{V} unchanged to retain the optimal rotation \mathbf{R} implies that the newly added term $\mathbf{U}^\top (\mathbf{p}_9 \mathbf{q}_9^\top) \mathbf{V}$ needs to be a diagonal matrix. Consequently, two supplementary pairs of GPs are further added to the cluster. Each pair of GPs is centroid symmetric so that they do not change the optimal translation. Similarly, we must ensure that all of these extra terms, which correspond to the inserted particles, form a diagonal matrix $\mathbf{\Lambda}^*$ satisfying:

$$\mathbf{U}^\top (\mathbf{p}_9 \mathbf{q}_9^\top + \mathbf{p}_{10} \mathbf{q}_{10}^\top + \mathbf{p}_{11} \mathbf{q}_{11}^\top) \mathbf{V} = \mathbf{\Lambda}^* \quad (11)$$

Since $\mathbf{p} = \mathbf{R} \mathbf{q} = \mathbf{U} \mathbf{V}^\top \mathbf{q}$, Eq. (11) can be simplified to

$$\mathbf{d} \mathbf{d}^\top + \mathbf{e} \mathbf{e}^\top + \mathbf{f} \mathbf{f}^\top = \mathbf{\Lambda}^* \quad (12)$$

where $\mathbf{d} = \mathbf{V}^\top \mathbf{q}_9$, $\mathbf{e} = \mathbf{V}^\top \mathbf{q}_{10}$, and $\mathbf{f} = \mathbf{V}^\top \mathbf{q}_{11}$. As all off-diagonal entries in $\mathbf{\Lambda}^*$ are zero, expanding Eq. (12) yields:

$$\begin{cases} d_1 d_2 + e_1 e_2 + f_1 f_2 &= 0 \\ d_1 d_3 + e_1 e_3 + f_1 f_3 &= 0 \\ d_2 d_3 + e_2 e_3 + f_2 f_3 &= 0 \end{cases} \quad (13)$$

Note that \mathbf{d} is known; we have six unknowns and only three equations. Thus, Eq. (13) is underconstrained and has multiple solutions. In our implementation, we just set $e_1 = d_1$, $e_2 = d_2$, $f_1 = d_3$, and solve for the other unknowns. In other words, SCPC adds six extra particles to keep $\mathbf{U}^\top \mathbf{A} \mathbf{V}$ diagonal as shown in Fig. 5.

4.3 Haptic rendering

The internal force at the HIP is computed as the derivative of the deformation energy with respect to its current position at the corresponding active

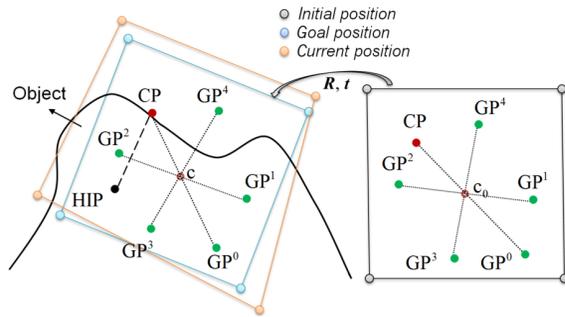


Fig. 5 Stable constraint particle coupling adds five GPs and one CP to the cluster.

bottom cluster:

$$\mathbf{f} = \kappa(\mathbf{g}_{cp} - \mathbf{x}_{cp}) \quad (14)$$

where κ is a constant determining the stiffness of the object. \mathbf{g}_{cp} is the goal position of CP. In addition, we apply virtual coupling [33] by adding a damping force to this framework when computing the coupling force, which guarantees stability of haptic rendering.

4.4 Visuo-haptic procedure

The visuo-haptic procedure includes a simulation thread and a haptic thread. In the simulation thread, a *bounded deformation tree* (BD-tree) [41–43] is constructed for collision detection.

The simulation thread executes the following sequentially at every time step:

1) **Initialization:** Set the proxy position \mathbf{x}_p as the device position of the last time step, and the the HIP position as \mathbf{x}_{HIP} . Set a line segment to be $\overline{\mathbf{x}_p \mathbf{x}_{HIP}}$.

2) **Collision handling:** If no collision was detected in the last time step, check whether a collision happens between $\overline{\mathbf{x}_p \mathbf{x}_{HIP}}$ and the object surface. If a collision was detected in the last time step, check whether a collision happens between $\overline{\mathbf{x}_{HIP} \mathbf{g}_{cp}}$ and the object surface. If there is an intersection with the surface mesh, the proxy position \mathbf{x}_p is set to this intersection point. The corresponding cluster of each level in the cluster hierarchy which contains the interaction point is labeled. If there is no collision, set the proxy position to the HIP position.

3) **Adding coupling particles:** Remove any coupling particles (CP and GPs) from the last time step. If a collision happens, add one CP and five GPs for cluster C_i using SCPC at the current level (as discussed in Section 4).

4) **Perform optimization:** Perform adaptive shape matching as in Algorithm 1.

5) **Surface mesh interpolation:** Update the positions of the surface mesh by trilinear interpolation.

The haptic thread executes the following:

1) **Contact force:** Compute contact forces as described in Section 4.3 using the current device state and constraint particles.

2) **Coupling force:** Compute coupling forces by virtual coupling based on the stability condition, and send the coupling forces to the device controller.

5 Experimental results

The proposed framework was implemented in Microsoft Visual C++ 2010 on a 64-bit Windows 7 PC equipped with an Intel Xeon 2.8 GHz CPU and 6.0 GB RAM. Only a single thread was used in our experiments and reported data. We refer readers to the accompanying videos in the Electronic Supplementary Material (ESM) for a more concrete visual impression of the results. Up to two Phantom Omni haptic devices [44] were used to interactively manipulate the deformable model shown in Fig. 6. Table 1 provides statistics of all the 3D models tested in our experiments while Table 2 shows detailed information about the cluster hierarchy setup as well as comparative time performances. As indicated in Table 2, the proposed adaptive iteration rule has a notable efficiency advantage over the classic HSM method.

Figure 7 compares our method and the naive shape

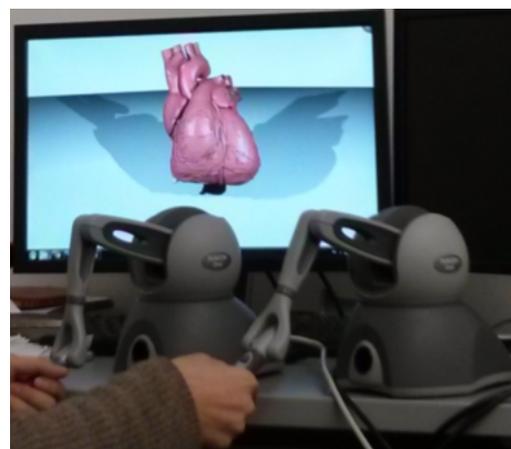


Fig. 6 Experimental setup.

Table 1 Model statistics. #Face and #Vert.: numbers of triangles and vertices of the input surface mesh, respectively. #Cluster: number of clusters at each level

Model	Statistics		Cluster hierarchy
	#Face	#Vert.	#Cluster
Bar	86	45	81/648/4000
Willow tree	5000	4000	703/3000/10,000
Elephant	85,000	42,000	202/954/5000
Sailboat	104,000	60,000	454/2000/12,000/ 71,000
Dinosaur	123,000	58,000	258/841/3000/ 17,000/98,000
Heart	53,000	178,000	914/6000/34,000

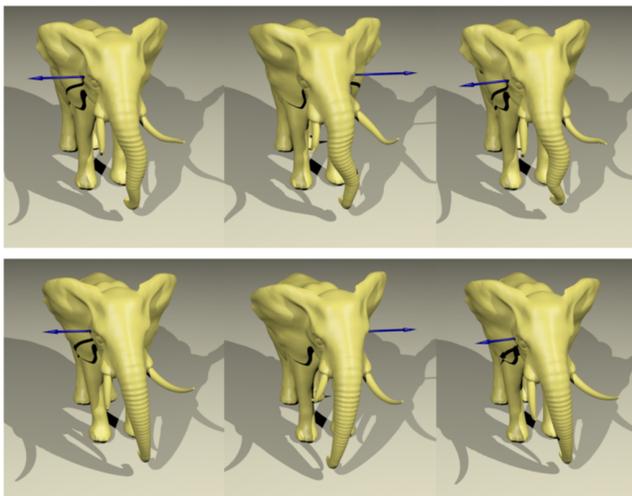


Fig. 7 Snapshots of simulation using the NSM method (top) and our method (bottom). Scripted forces are highlighted by blue arrows in the figure.

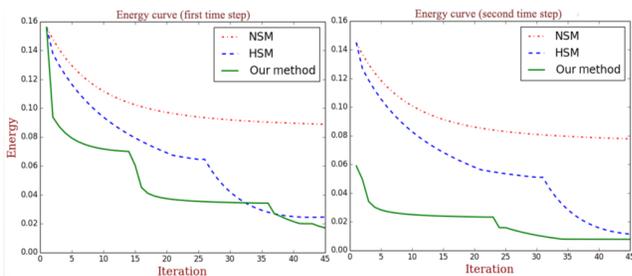


Fig. 8 Energy changes during iterations of the first two time steps for the elephant model in Fig. 7.

matching (NSM) approach. The forces, indicated as blue arrows in the figure, are applied to shake the trunk of the elephant. At each time step, the surface mesh is updated after each cluster completes just one shape matching. Using NSM (top row), we can clearly see an unnatural wave-like deformation in the nose. With our method, a more natural result is produced (bottom row). This result also suggests

the slow energy convergence of the NSM method. To achieve the same energy reduction as the one after 15 iterations of our adaptive method, NSM needs over 2200 iterations. In terms of computational efficiency, classic HSM is about 190 times faster than NSM while our method is up to 310 times faster. Table 2 reports the simulation performance in detail. On average, our method is orders-of-magnitude faster than NSM and consistently outperforms HSM by 30%–50%. In some extreme cases (e.g., very large and subtle deformations occur), our method provides better performance improvements due to the adaptive iteration strategies used. Figure 8 shows how the deformation potential is reduced along the simulation during the first two time steps using NSM, HSM, and our method, for the elephant animation shown in Fig. 7.

Figure 4 compares results of collision handling using CPC and SCPC. In this experiment, the user slightly pushes the bar model along the negative z axis with an HIP. CPC results in unexpected bending because it does not preserve the optimal rotation for the corresponding cluster. If the HIP leaves the object immediately, such bending is undone. As a result, the model jumps back and forth between these two different deformation configurations. Our SCPC method eliminates such artifacts and produces smooth deformation instead.

The force rendering in our proposed framework is smooth and natural. In Fig. 9(a), the user bends the bar model with a single haptic device; the corresponding force variation is shown. The harder the bar is bent the larger force rendered, which provides a reasonable interaction to the user. In

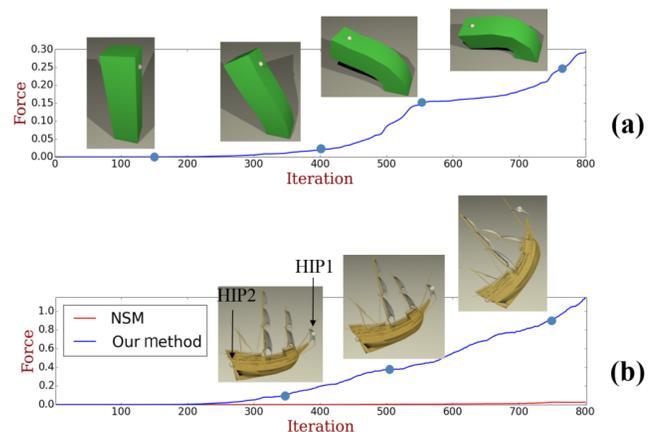


Fig. 9 Force rendered at HIP along with the SM iteration with (a) one and (b) two haptic devices.

Table 2 Time and computation performance. Comp. intensity: average number of SM iterations required at each step to achieve energy convergence. Time benchmark: average computational time for a single time step when simulating the 3D model

Model	Comp. intensity (#SM iteration)			Time benchmark		
	NSM	HSM	Our method	NSM	HSM	Our method
Bar	2.1×10^6	9000	6000 ($\sim 352\times$)	1.07 s	4 ms	3 ms ($\sim 417\times$)
Willow tree	10.9×10^6	35,000	29,000 ($\sim 372\times$)	3.81 s	16 ms	11 ms ($\sim 354\times$)
Elephant	6.9×10^6	27,000	22,000 ($\sim 314\times$)	1.86 s	10 ms	6 ms ($\sim 310\times$)
Sailboat	34×10^6	82,000	57,000 ($\sim 594\times$)	8.41 s	26 ms	12 ms ($\sim 675\times$)
Dinosaur	48×10^6	66,000	59,000 ($\sim 822\times$)	12.60 s	22 ms	16 ms ($\sim 788\times$)
Heart	19×10^6	58	48,000 ($\sim 407\times$)	5.10 s	18 ms	12 ms ($\sim 407\times$)

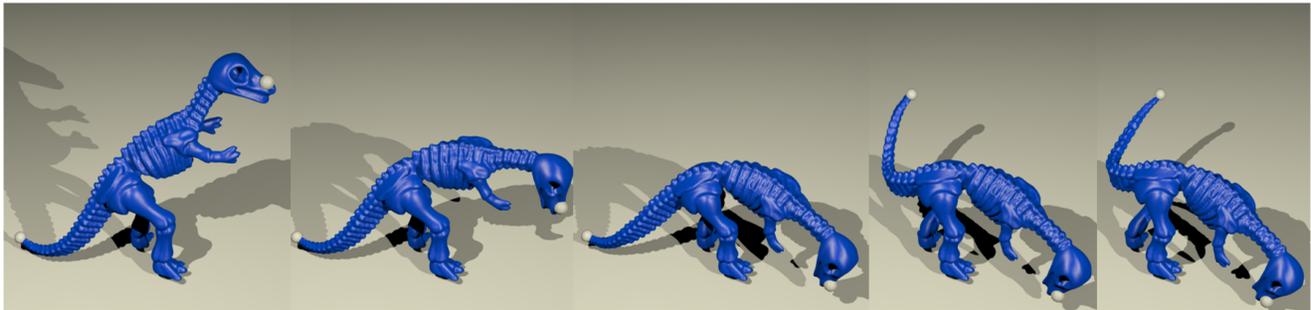
**Fig. 10** Willow tree swaying in the wind. We impose light, medium, and strong wind fields to the model; our adaptive iteration strategy significantly reduces the number of iterations used. See the video in the ESM for more details.**Fig. 11** Two haptic devices interact with a deformable dinosaur model. HIPs are shown as grey spheres. Realistic results are produced even under extreme deformation. A five-level cluster hierarchy is used in this example.

Fig. 9(b), two haptic devices are used. HIP1 is attached to the prow of the sailboat while HIP2 is attached to the stern. HIP2 is held fixed by one user while the other user pulls HIP1 up. We record the magnitude of the rendered force for HIP2 during SM iteration, using both NSM and our method. It can be clearly seen that, thanks to the accelerated energy convergence, HIP2 instantly renders the responsive force due to the interaction with HIP1 while NSM suffers a lengthy force delay.

Our method is particularly good at simulating geometrically complex models with rich local details. Figure 10 shows the snapshots of a willow tree swaying in the wind. We refer readers to the accompanying video in the ESM for details, where we

show three different scenarios with weak, medium, and strong winds. The proposed adaptive simulation strategy is able to well accommodate wind fields of different intensities, and natural results are produced yet the simulation is still efficient.

In Fig. 11, two haptic devices interact with the dinosaur model simultaneously. It can be seen that our system is quite robust even under extreme deformations imposed by the user. Another example is shown in Fig. 12. In this case we assess our framework using some medical data. It is a non-manifold mesh with ill-positioned triangles and edge topology. Our system can still create a realistic virtual environment allowing the user to interactively manipulate the model using multiple haptic devices.

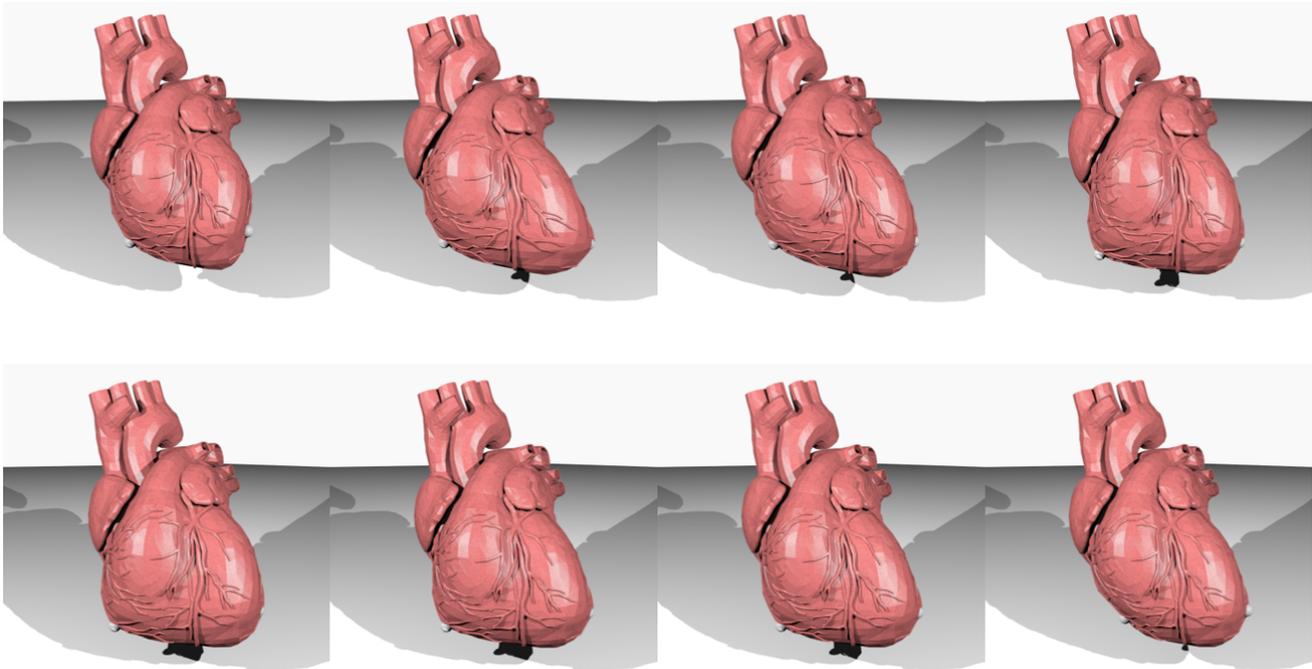


Fig. 12 A beating heart model is interactively manipulated by two haptic devices. The HIPs are shown as grey spheres. A three-level cluster hierarchy is used in this example.

6 Conclusions and future work

In this paper, we have presented a novel system to simulate deformation of geometrically complex objects with real-time interaction using multiple haptic devices based on adaptive hierarchical shape matching. We use a multi-resolution hierarchy of particle clouds with three adaptive strategies to boost the energy convergence speed while still well capturing locally detailed deformation. Our approach works well with existing methods such as FLSM. Multi-haptic interaction is a challenging problem and it is well handled with our new particle coupling technique. It guarantees optimality of the cluster's existing rotation and translation, and preserves a smooth deformation trajectory. Based on this technique, the force rendered by the haptic device is smooth and realistic.

There are many possibilities for future work that could improve the current system. First of all, the computation of the optimal rotation and translation is essentially independent for each cluster. Thus, a parallel implementation of the proposed system could give much better performance and enable users to interact with more complicated scenarios in real time. Secondly, shape matching depends

on a geometry-based pseudo-dynamic deformable model. We still lack of a good representation to accurately incorporate elastic materials with different parameters such as Young's modulus or Poisson's ratio. Because of this limitation, the proposed system is not yet suitable for applications that require highly accurate simulation, such as optimization for 3D printing. How to integrate material properties into the system remains an interesting and challenging topic. Another promising direction is to further extend our system to a networked tele-immersive virtual environment. In this case, special care needs to be taken to handle connection stability of the network and provide a high-quality user experience.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1012975. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which

permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Electronic Supplementary Material Supplementary materials are available in the online version of this article at <http://dx.doi.org/10.1007/s41095-015-0023-3>.

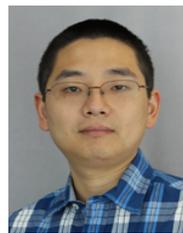
References

- [1] Barbič, J.; James, D. L. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics* Vol. 1, No. 1, 39–52, 2008.
- [2] Tian, Y.; Yang, Y.; Guo, X.; Prabhakaran, B. Haptic-enabled interactive rendering of deformable objects based on shape matching. In: Proceedings of 2013 IEEE International Symposium on Haptic Audio Visual Environments and Games, 75–80, 2013.
- [3] Ullrich, S.; Kuhlen, T. Haptic palpation for medical simulation in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* Vol. 18, No. 4, 617–625, 2012.
- [4] Lim, Y.-J.; De, S. Nonlinear tissue response modeling for physically realistic virtual surgery using paff. In: Proceedings of Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint, 479–480, 2005.
- [5] Ye, Y.; Liu, P. X. Improving haptic feedback fidelity in wave-variablebased teleoperation orientated to telemedical applications. *IEEE Transactions on Instrumentation and Measurement* Vol. 58, No. 8, 2847–2855, 2009.
- [6] Xu, S.; Liu, X. P.; Zhang, H.; Hu, L. A nonlinear viscoelastic tensor-mass visual model for surgery simulation. *IEEE Transactions on Instrumentation and Measurement* Vol. 60, No. 1, 14–20, 2011.
- [7] Alamri, A.; Eid, M.; Iglesias, R.; Shirmohammadi, S.; El-Saddik, A. Haptic virtual rehabilitation exercises for poststroke diagnosis. *IEEE Transactions on Instrumentation and Measurement* Vol. 57, No. 9, 1876–1884, 2008.
- [8] Müller, M.; Heidelberger, B.; Teschner, M.; Gross, M. Meshless deformations based on shape matching. *ACM Transactions on Graphics* Vol. 24, No. 3, 471–478, 2005.
- [9] Pentland, A.; Williams, J. Good vibrations: Modal dynamics for graphics and animation. In: Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques, 215–222, 1989.
- [10] Choi, M. G.; Ko, H.-S. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Transactions on Visualization and Computer Graphics* Vol. 11, No. 1, 91–101, 2005.
- [11] Barbič J.; James, D. L. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics* Vol. 24, No. 3, 982–990, 2005.
- [12] Yang, Y.; Rong, G.; Torres, L.; Guo, X. Real-time hybrid solid simulation: Spectral unification of deformable and rigid materials. *Computer Animation and Virtual Worlds* Vol. 21, Nos. 3–4, 151–159, 2010.
- [13] Galoppo, N.; Tekin, S.; Otaduy, M. A.; Gross, M.; Lin, M. C. Interactive haptic rendering of high-resolution deformable objects. In: *Lecture Notes in Computer Science, Vol. 4563*. Berlin Heidelberg: Springer, 215–223, 2007.
- [14] An, S. S.; Kim, T.; James, D. L. Optimizing cubature for efficient integration of subspace deformations. In: Proceedings of ACM SIGGRAPH Asia 2008 papers, Article No. 165, 2008.
- [15] Tang, Z.; Yang, Y.; Guo, X.; Prabhakaran, B. Distributed haptic interactions with physically based 3d deformable models over lossy networks. *IEEE Transactions on Haptics* Vol. 6, No. 4, 417–428, 2013.
- [16] Irving, G.; Teran, J.; Fedkiw, R. Invertible finite elements for robust simulation of large deformation. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 131–140, 2004.
- [17] Teschner, M.; Heidelberger, B.; Müller, M.; Gross, M. A versatile and robust model for geometrically complex deformable solids. In: Proceedings of the Computer Graphics International, 312–319, 2004.
- [18] McAdams, A.; Zhu, Y.; Selle, A.; Empey, M.; Tamstorf, R.; Teran, J.; Sifakis, E. Efficient elasticity for character skinning with contact and collisions. *ACM Transactions on Graphics* Vol. 30, No. 4, Article No. 37, 2011.
- [19] Liu, T.; Bargteil, A. W.; O'Brien, J. F.; Kavan, L. Fast simulation of mass-spring systems. *ACM Transactions on Graphics* Vol. 32, No. 6, Article No. 214, 2013.
- [20] Bender, J.; Müller, M.; Otaduy, M. A.; Teschner, M. Position-based methods for the simulation of solid objects in computer graphics. In: Proceedings of EUROGRAPHICS 2013 State of the Art Reports, 2013.
- [21] Müller, M.; Heidelberger, B.; Hennix, M.; Ratcliff, J. Position based dynamics. *Journal of Visual Communication and Image Representation* Vol. 18, No. 2, 109–118, 2007.
- [22] Müller, M.; Chentanez, N. Solid simulation with oriented particles. *ACM Transactions on Graphics* Vol. 30, No. 4, Article No. 92, 2011.
- [23] Rivers, A. R.; James, D. L. FastLSM: Fast lattice shape matching for robust real-time deformation. In: Proceedings of ACM SIGGRAPH 2007 papers, Article No. 82, 2007.
- [24] Steinemann, D.; Otaduy, M. A.; Gross, M. Fast adaptive shape matching deformations. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 87–94, 2008.
- [25] Botsch, M.; Pauly, M.; Gross, M.; Kobbelt, L. PriMo:

- Coupled prisms for intuitive surface modeling. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, 11–20, 2006.
- [26] Wang, Y.; Xiong, Y.; Xu, K.; Tan, K.; Guo, G. A mass-spring model for surface mesh deformation based on shape matching. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, 375–380, 2006.
- [27] Tian, Y.; Yang, Y.; Guo, X.; Prabhakaran, B. A multigrid approach for bandwidth and display resolution aware streaming of 3D deformations. In: Proceedings of the 21st ACM International Conference on Multimedia, 693–696, 2013.
- [28] Bender, J.; Weber, D.; Diziol, R. Fast and stable cloth simulation based on multi-resolution shape matching. *Computers & Graphics* Vol. 37, No. 8, 945–954, 2013.
- [29] Müller, M. Hierarchical position based dynamics. In: Proceedings of Virtual Reality Interactions and Physical Simulations 2008. Available at <http://matthias-mueller-fischer.ch/publications/hpbd.pdf>.
- [30] Otaduy, M. A.; Lin, M. C. Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration. In: Proceedings of Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint, 247–256, 2005.
- [31] Bro-Nielsen, M.; Cotin, S. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum* Vol. 15, No. 3, 57–66, 1996.
- [32] Martin, S.; Kaufmann, P.; Botsch, M.; Grinspun, E.; Gross, M. Unified simulation of elastic rods, shells, and solids. In: Proceedings of ACM SIGGRAPH 2010 Papers, Article No. 39, 2010.
- [33] Colgate, J. E.; Brown, J. M. Factors affecting the ZWidth of a haptic display. In: Proceedings of 1994 IEEE International Conference on Robotics and Automation, Vol. 4, 3205–3210, 1994.
- [34] Colgate, J. E.; Stanley, M. C.; Brown, J. M. Issues in the haptic display of tool use. In: Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, 140–145, 1995.
- [35] Wan, M.; McNeely, W. A. Quasistatic approach approximation for 6 degrees-of-freedom haptic rendering. In: Proceedings of IEEE Visualization, 257–262, 2003.
- [36] Bouaziz, S.; Martin, S.; Liu, T.; Kavan, L.; Pauly, M. Projective dynamics: Fusing constraint projections for fast simulation. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 154, 2014.
- [37] Briggs, W. L.; Henson, V. E.; McCormick, S. F. *A Multigrid Tutorial*, 2nd end. Philadelphia, PA, USA: SIAM: Society for Industrial and Applied Mathematics, 2000.
- [38] Zhu, Y.; Sifakis, E.; Teran, J.; Brandt, A. An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Transactions on Graphics* Vol. 29, No. 2, Article No. 16, 2010.
- [39] Shoemake, K. Animating rotation with quaternion curves. *ACM SIGGRAPH Computer Graphics* Vol. 19, No. 3, 245–254, 1985.
- [40] Jiménez, P.; Thomas, F.; Torras, C. 3D collision detection: A survey. *Computers & Graphics* Vol. 25, No. 2, 269–285, 2001.
- [41] Quinlan, S. Efficient distance computation between non-convex objects. In: Proceedings of 1994 IEEE International Conference on Robotics and Automation, Vol. 4, 3324–3329, 1994.
- [42] Hubbard, P. M. Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics* Vol. 1, No. 3, 218–230, 1995.
- [43] Bradshaw, G.; O’Sullivan, C. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics* Vol. 23, No. 1, 1–26, 2004.
- [44] Phantom omni haptic.



Yuan Tian is a Ph.D. candidate in the Computer Science Department, University of Texas at Dallas, Richardson, USA. His research interests include physics-based simulation and modeling, haptic rendering, and tele-immersive system.



Yin Yang received his Ph.D. degree in computer science from the University of Texas at Dallas in 2013. He is an assistant professor in the Electrical Communication Engineering Department, University of New Mexico, Albuquerque, USA. His research interests include physics-based animation and simulation and related applications, scientific visualization, and medical imaging analysis.



Xiaohu Guo received his Ph.D. degree in computer science from the State University of New York at Stony Brook in 2006. He is an associate professor of computer science at the University of Texas at Dallas. His research interests include computer graphics, animation and visualization, with an emphasis on geometric- and physics-based modeling, spectral geometric analysis, deformable models, centroidal Voronoi tessellation, GPU algorithms, and 3D and 4D medical image analysis. He received the prestigious US National Science Foundation CAREER Award in 2012. He is a member of the IEEE.



Balakrishnan Prabhakaran received his Ph.D. degree in computer science from the Indian Institute of Technology, Madras, India, in 1995. He is currently a professor of computer science in the University of Texas at Dallas. He has been working in the area of multimedia systems: animation and multimedia databases, authoring and presentation, resource management, and scalable web-based multimedia presentation servers. He received the US National Science Foundation CAREER Award in 2003. He was the general co-chair of ACM Multimedia 2011 and has served as an associate chair of the ACM Multimedia Conferences in 2006, 2003, 2000, and 1999. He has served as the guest editor of special issues on various topics for different

multimedia journals. He also serves on the editorial boards of journals such as *Multimedia Systems* (Springer), *Multimedia Tools and Applications* (Springer), *Journal of Multimedia* (Academy Publisher), and *International Journal of Multimedia Data Engineering and Management* (Information Resources Management Association). He is also the Editor-in-Chief of the *ACM Special Interest Group on Multimedia* (SIGMM) online magazine.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.